LAOC II – Atividade 3

Nome: Gustavo de Assis Xavier

Esse trabalho consiste na análise dos gráficos de dois testes que se baseiam em gerar e pesquisar em árvores binárias balanceadas, com elementos ordenados e aleatórios.

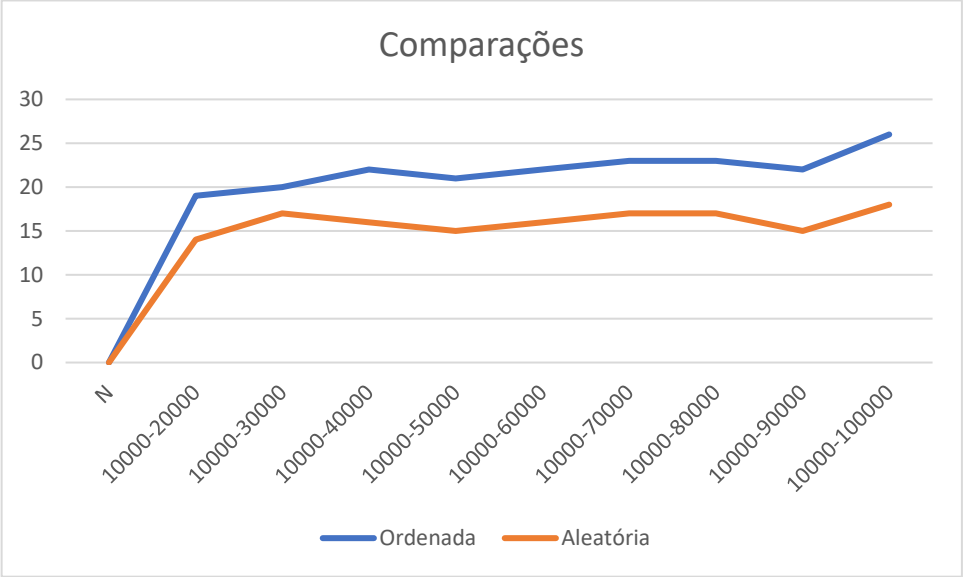
Os testes consistem em criar diversas árvores com N elementos que variam de 10000 a 100000, divididos em intervalos de 10000, ou seja, a primeira árvore contém elementos de 10000 a 20000, a segunda árvore possui elementos de 10000 a 30000, e assim por diante.

Para aumentar a fidelidade dos dados, o evento de pesquisa foi feito 100 vezes para cada caso, e disso, obtivemos as médias que serão apresentadas nas tabelas e gráficos a seguir.

Os resultados obtidos dos testes na árvore com balanceamento são apresentados pela tabela abaixo:

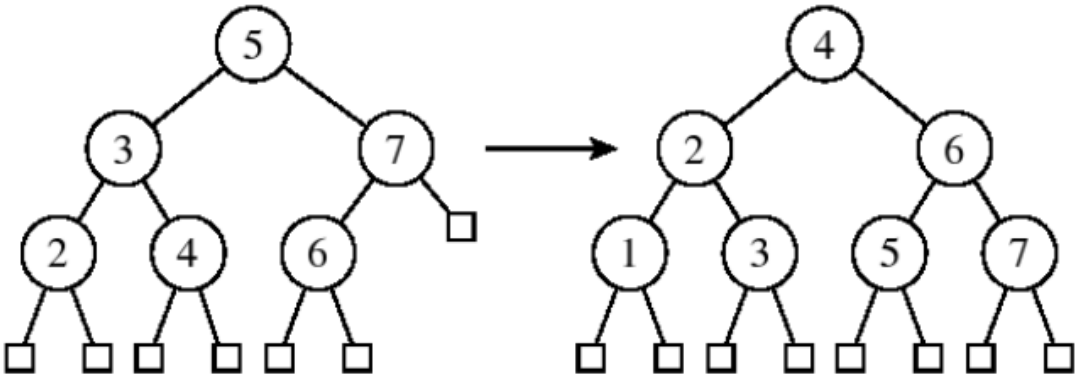
	Árvore balanceada			
	Ordenada		Aleatória	
N	Comparações	Tempo (ns)	Comparações	Tempo (ns)
10000-20000	19	2809	14	448
10000-30000	20	1909	17	90
10000-40000	22	2815	16	280
10000-50000	21	226	15	70
10000-60000	22	199	16	107
10000-70000	23	160	17	81
10000-80000	23	123	17	89
10000-90000	22	91	15	124
10000-100000	26	105	18	84

A seguir a representação gráfica desses dados:

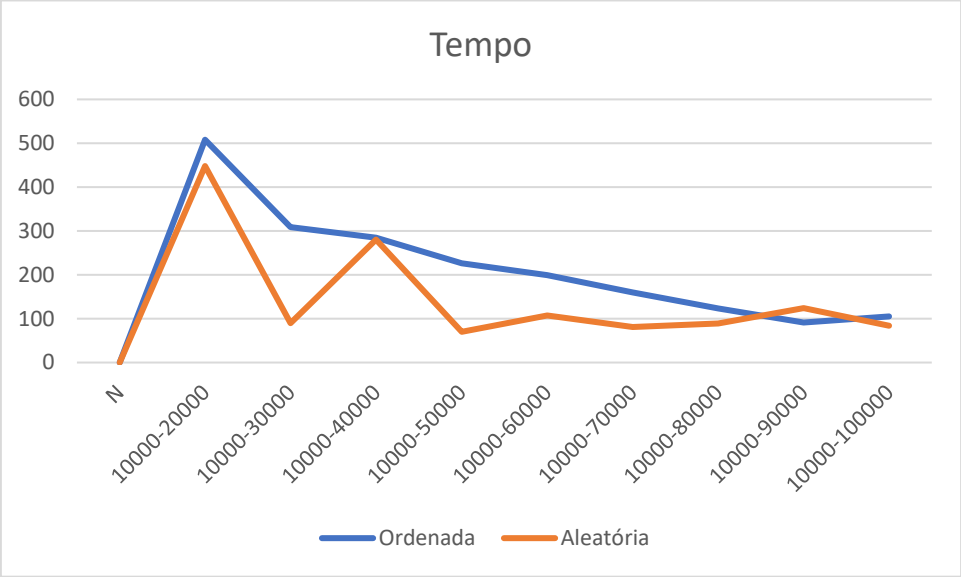


Podemos observar que a diferença é mínima entre a busca de um elemento inexistente na árvore criada a partir de elementos ordenados e a árvore binária criada a partir de elementos aleatórios. Isso se dá ao fato de a árvore com balanceamento minimizar tempo médio de pesquisa para uma distribuição uniforme das chaves, onde cada chave é igualmente provável de ser usada em uma pesquisa.

Em outras palavras, após cada inserção na árvore balanceada, há um balanceamento entre os ramos da árvore, o que reduz o tempo de pesquisa de um elemento nela.



Já com relação ao tempo, temos que:



Como esperado, a diferença de tempo entre as duas abordagens foi pequena, reflexo da pequena diferença entre os números de comparações. Ainda assim, é importante ressaltar que o tempo não é uma medida tão eficaz, não nesse caso, devido às múltiplas variáveis que podem afetá-la, como programas em segundo plano e/ou otimizações que o compilador faz no código.

Agora, levando em consideração os dados da árvore sem balanceamento:

	Sem balanceamento			
	Ordenada		Aleatória	
N	Comparações	Tempo	Comparações	Tempo
1000-2000	1001	25235	9	148
1000-3000	2001	12490	8	197
1000-4000	3001	19462	8	180
1000-5000	4001	29161	10	182
1000-6000	5001	31626	7	161
1000-7000	6001	52107	12	162
1000-8000	7001	42393	11	162
1000-9000	8001	57087	16	360

A comparação gráfica desses dados não faz sentido devido a eles se basearem em intervalos numéricos diferentes. No entanto, é possível notar que mesmo que a árvore com balanceamento abranja um intervalo numérico muito maior, seus resultados são mais

satisfatórios (tanto na busca aleatória como na ordenada) com destaque para busca em uma árvore na qual a inserção foi feita de forma ordenada, isso devido a árvore binaria sem balanceamento ordenada se comportar como uma lista linear. Com relação a busca nas árvores com inserção aleatória, não sentimentos tanta diferença entre a árvore com balanceamento e a sem balanceamento, isso por que a árvore binaria sem balanceamento com itens aleatórios é, em média, apenas 39% pior que uma árvore completamente balanceada.